ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ТЕЛЕКОММУНИКАЦИИ
СПбГУТ))) Декабрь 2019 Том 7 № 4 ISSN 2307-1303 http://www.itt.sut.ru
TELECOM IT
December 2019 Vol. 7 Iss. 4 ISSN 2307-1303 http://www.itt.sut.ru

ПРИМЕНЕНИЕ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ПОИСКА УЯЗВИМОСТЕЙ В ПРОГРАММНОМ КОДЕ

М. В. Буйневич[1,2*], П. Е. Жуковская[1], К. Е. Израилов[1,3], В. В. Покусов[2,4]

[1]Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича, Санкт-Петербург, 193232, Российская Федерация
[2]Санкт-Петербургский университет государственной противопожарной службы МЧС России, Санкт-Петербург, 196105, Российская Федерация
[3]Санкт-Петербургский институт информатики и автоматизации Российской академии наук, Санкт-Петербург, 199178, Российская Федерация
[4]Казахстанская Ассоциация информационной безопасности г. Алматы, A05F2K5, Республика Казахстан

*Адрес для переписки: bmv1958@yandex.ru

Аннотация
В статье рассмотрена возможность применения искусственного интеллекта в области информационной безопасности. Для этого выделено следующее противоречие предметной области: заточенность алгоритмов обнаружения под конкретные уязвимости в коде VS постоянная модификация кода уязвимостей. Для его разрешения рассматривается объект – программное обеспечение с уязвимостями в процессе его разработки, на предмет – способов интеллектуального анализа его характеристик. Предложено гипотетическое решение путем применения Технологии машинного обучения для выявления уязвимостей в программном обеспечении, используемое огромный объем накопленных знаний. Также, приведены возможные признаки представлений программного обеспечения, используемые для работы Технологии.

Ключевые слова
Машинное обучение, уязвимости, ошибки, дефекты, программное обеспечение, машинный код, исходный код.

Информация о статье
УДК 004.49
Язык статьи – русский.
Поступила в редакцию 04.04.2019, принята к печати 30.12.19.

Ссылка для цитирования: Буйневич М. В., Жуковская П. Е., Израилов К. Е., Покусов В. В. Применение машинного обучения для поиска уязвимостей в программном коде // Информационные технологии и телекоммуникации. 2019. Том 7. № 4. С. 59–65. DOI 10.31854/2307-1303-2019-7-4-59-65.

APPLICATION OF MACHINE TRAINING TO SEARCH VULNERABILITIES IN THE SOFTWARE CODE

M. Buinevich[1,2*], P. Zhukovskaya[1], K. Izrailov[1,3], V. Pokussov[2,4]

[1]The Bonch-Bruevich Saint-Petersburg State University of Telecommunications,
St. Petersburg, 193232, Russian Federation
[2]Saint-Petersburg University of State Fire Service of EMERCOM of Russia,
St. Petersburg, 196105, Russian Federation
[3] St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences,
St. Petersburg, 199178, Russian Federation
[4] Kazakhstanskaya Assotsiatsiya informatsionnoy bezopasnosti
Almaty, A05F2K5, Kazakhstan Republic

*Corresponding author: bmv1958@yandex.ru

Abstract—The article considers the possibility of using artificial intelligence in the field of information security. For this, the following domain contradiction was highlighted: sharpening of detection algorithms for specific vulnerabilities in VS code, constant modification of the vulnerability code. To resolve the contradiction, an object is considered – software with vulnerabilities in the process of its development, for the subject – ways of intellectual analysis of its characteristics. A hypothetical solution is proposed through the use of Machine Learning Technology to identify vulnerabilities in software, using a huge amount of accumulated knowledge. Also, possible signs of software representations used for the operation of the Technology are given.

Keywords—Machine learning, vulnerabilities, errors, defects, software, machine code, source code.

Article info
Article in Russian.
Received 04.04.2019, accepted 30.12.19.

For citation: Buinevich M., Zhukovskaya P., Izrailov K., Pokussov V.: Application of Machine Training to Search Vulnerabilities in the Software Code // Telecom IT. 2019. Vol. 7. Iss. 4. pp. 59–65 (in Russian). DOI 10.31854/2307-1303-2019-7-4-59-65.

Введение

Проблема обнаружения уязвимостей в программном обеспечении (ПО) появилась в тот же момент, когда появилось и само ПО. И несмотря на постоянную актуальность проблемы, огромный накопленный теоретический и практический опыт, существенные риски от использования уязвимостей как для бизнеса, так для общества и государства, сама проблема еще далека от ее полного разрешения [1]. Ситуация осложняется еще и тем, что информатизация, проникнув во все сферы жизнедеятельности, помимо инновационных достижений принесла также и множество качественно новых информационных угроз [2].

Одной из причин такого положения дел можно считать неразрешенность следующего противоречия. С одной стороны, для борьбы с конкретными уязвимостями применяются соответствующие алгоритмы их обнаружения [3]. При этом, исходя из субъективности понятия «уязвимость», в основу алгоритмов

должен быть заложен опыт экспертов [4, 5]. С другой стороны, уязвимости достаточно быстро и просто модифицируемы, что не позволяет применять «точные» и устаревающие правила [6].

Разрешение противоречия может лежать в областях Больших данных и Технологии машинного обучения (ТМО); так, первая сможет учесть огромный накопленный практический опыт экспертов, а вторая – построить на базе этого опыта алгоритмы обнаружения уязвимостей с учетом видоизменений последних (по аналогии с [7, 8]). Далее в статье будет кратко раскрыта методология применения ТМО для поиска уязвимостей в ПО.

Виды и задачи Технологии машинного обучения

ТМО представляет собой способ предсказания результата по входным данным; при этом, она особо эффективна там, где использование экспертов для решения этой задачи считается крайне трудоемким. Идея применения ТМО к ПО заключается в его интеллектуальном анализе на предмет наличия уязвимостей. Рассмотрим более подробно виды ТМО и основные решаемые ей задачи для обоснования их применимости в интересах поиска уязвимостей в ПО.

Все виды систем, реализующих ТМО, можно условно поделить на два: обучение с учителем и без него [9]. К первому виду относятся системы, которые обучаются экспериментатором с помощью набора примеров типа «вход-выход» – *обучающей выборки*; зависимость между «входом» и «выходом» и должна быть выявлена (например, распознавание изображений). Используя ее, система сможет определять «выходы» для новых «входов», на которые она не была обучена ранее. Ко второму виду относятся системы, которые не требуют вмешательства со стороны экспериментатора – то есть система сама должна выявить внутренние закономерности между объектами данных (например, кластеризовать объекты).

Типичными задачами, решаемыми ТМО, являются следующие:

- классификация – отнесение объекта к некоторой категории по набору его признаков, как правило в виде ответа – «да/нет» (например, разделение изображений дорожных знаков на предупреждающие и запрещающие);
- восстановление регрессии – прогнозирование характеристики объекта по его признакам (например, предсказание курса Bitcoin/USD на криптовалютной бирже);
- понижение размерности – уменьшение числа признаков данных (например, для упрощения визуализации, экономии выделяемого количества памяти, ускорения работы других алгоритмов и т. п.);
- кластеризация – разделение объектов на группы (например, выделение подпрограмм, классов и модулей в машинном коде);
- выявление аномалий – обнаружение выбросов в наборе данных, для которых скорее всего нет (в отличии от задачи классификации) обучающих примеров (например, детектирование сетевых атак).

Поиск уязвимостей в коде

Исходя из видов ТМО и решаемых ею задач, одной из наиболее подходящих для обнаружения уязвимостей является задача классификации с помощью обучения с учителем – путем соответствующей реализации программно-аппаратной системы (далее – Система). При этом, входными данными может быть описание

ПО в одном из его Представлений (некой стадии состояния ПО, которые будут описаны далее), обучающей выборкой будет множество известных и искусственно сгенерированных уязвимостей в этом Представлении, признаками – особенности ПО в Представлении, а результатом – класс уязвимости или просто факт ее наличия. Исходя из того, что даже размер исходного кода одной функции ПО может колебаться от одной строки до тысячи, то предлагаемая Система должна анализировать код частями – путем перемещения некоторого окна от начала и до конца каждой функции по всей совокупности файлов исходного кода; также, возможно будет оправданным обнаружение каждой из уязвимостей по отдельности, используя размер перемещаемого окна, равный максимальной длине уязвимости данного класса.

Одним из возможных способов деления жизненного цикла ПО является авторский, заключающийся в прохождения программой следующей последовательности его Представлений [10, 11]:

Идея – некий идеализированный вид ПО, к которому его конечная реализация должна стремиться.

Концептуальная модель – описание ПО в базовых терминах и их взаимосвязях.

Архитектура – деление ПО на модули, выбор технологий будущей разработки и т. п.

Алгоритмы – пошаговое описание работы подпрограмм модулей.

Исходный код – код реализации алгоритмов на заданном языке программирования.

Ассемблерный код – описание алгоритмов на языке инструкций процессора.

Машинный код – бинарный код ПО, готовый для непосредственного выполнения процессором.

Необходимо отметить, что здесь под ПО понимается классическое, разрабатываемое на императивном языке программирования и выполняемое на реальном процессоре.

Исходя из вышеизложенного, а также содержащегося в предыдущих исследованиях, в каждом Представлении возможно возникновение своего типа уязвимостей [12, 13, 14] и, следовательно, необходимо создание своей обучающей выборки для Системы. Таким образом, Система может анализировать ПО на каждом из Представлений на предмет наличия уязвимостей. Приведем гипотетические признаки ПО в каждом из Представлений, которые могут быть использованы для ТМО [15]:

Идея – изначально считается, что уязвимости отсутствуют.

Концептуальная модель – характеристики базовых элементов и их взаимосвязь.

Архитектура – перечень используемых технологий и потоки управления/данных между модулями.

Алгоритмы – свойства узлов блок-схемы.

Исходный код – последовательность синтаксически подобных лексем или некоторые его метрики [16].

Ассемблерный код – семантика инструкций процессора [17].

Машинный код – N-граммы последовательности байт [18].

Для последних четырех Представлений в качестве признаков может быть также взята топология графа потока управления и набор вызовов внешних функций (то есть используемый API).

Заключение

Исходя из накопившейся Best Practice ПО, существующих наработок в данной области, а также возможностей ТМО, применение последней для обнаружения уязвимостей может оказаться крайне востребованным. Дальнейшим этапом исследования должна стать практическая реализация предлагаемой Системы, оценка ее результативности, выбор наилучших алгоритмов, признаков и параметров машинного обучения, а также проверка Системы на реальном ПО во всех его Представлениях.

Литература

1. Израилов К. Е. Анализ состояния в области безопасности программного обеспечения // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО-2013): сборник научных статей II Международной научно-технической и научно-методической конференции. 2013. С. 874–877.

2. Буйневич М. В., Израилов К. Е., Мостович Д. И., Ярошенко А. Ю. Проблемные вопросы нейтрализации уязвимостей программного кода телекоммуникационных устройств // Проблемы управления рисками в техносфере. 2016. № 3 (39). С. 81–89.

3. Буйневич М. В., Васильева И. Н., Воробьев Т. М., Гниденко И. Г., Егорова И. В., Еникеева Л. А. и др. Защита информации в компьютерных системах: монография. – СПб.: СПГЭУ, 2017. 163 с.

4. Буйневич М. В., Израилов К. Е., Мостович Д. И. Сравнительный анализ подходов к поиску уязвимостей в программном коде // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО-2016): сборник научных статей V Международной научно-технической и научно-методической конференции. 2016. Т. 1. С. 256–260.

5. Израилов К. Е. Поиск уязвимостей в различных представлениях машинного кода // Информационная безопасность регионов России (ИБРР-2015): материалы IX Санкт-Петербургской межрегиональной конференции. 2015. С. 157.

6. Израилов К. Е., Гололобов Н. В., Краскин Г. А. Метод анализа вредоносного программного обеспечения на базе Fuzzy Hash // Информатизация и связь. 2019. № 2. С. 36–44.

7. Израилов К. Е. Модель прогнозирования угроз телекоммуникационной системы на базе искусственной нейронной сети // Вестник ИНЖЭКОНа. Серия: Технические науки. 2012. № 8(59). С. 150–153.

8. Бирюков А. А., Израилов К. Е. Сравнительный анализ моделей угроз информационной безопасности в интересах применимости для многоэтапных схем атак // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО-2017): сборник научных статей VI Международной научно-технической и научно-методической конференции. 2017. С. 108–112.

9. Балуева А. В., Десницкий В. А. Подход к обнаружению атак типа denial-of-sleep в киберфизических системах на основе методов машинного обучения // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2019): сборник научных статей VIII Международной научно-технической и научно-методической конференции. 2019. С. 97–103.

10. Buinevich M., Izrailov K., Vladyko A. The life cycle of vulnerabilities in the representations of software for telecommunication devices // 18th International Conference On Advanced Communications Technology (ICACT-2016). 2016. pp. 430–435.

11. Buinevich M., Izrailov K., Vladyko A. Metric of vulnerability at the base of the life cycle of software representations // 20th International Conference on Advanced Communication Technology (ICACT 2018). 2018. pp. 1–8.

12. Израилов К. Е. Архитектурные уязвимости программного обеспечения // Шестой научный конгресс студентов и аспирантов СПбГИЭУ (ИНЖЭКОН-2013): сборник тезисов докладов научно-практической конференции факультета информационных систем в экономике и управлении «Инфокоммуникационные технологии и математические методы». 2013. С. 35.

13. Буйневич М. В., Израилов К. Е., Щербаков О. В. Структурная модель машинного кода, специализированная для поиска уязвимостей в программном обеспечении автоматизированных систем управления // Проблемы управления рисками в техносфере. 2014. № 3 (31). С. 68–74.

14. Буйневич М. В., Израилов К. Е., Щербаков О. В. Модель машинного кода, специализированная для поиска уязвимостей // Вестник Воронежского института ГПС МЧС России. 2014. № 2 (11). С. 46–51.

15. Израилов К. Е. Рассмотрение представлений кода программ с позиций метаданных // Фундаментальные исследования и инновации в национальных исследовательских университетах: материалы Всероссийской научно-методической конференции. 2012. Т. 5. С. 176–180.

16. Белов Ю. С., Юхименко Н. В. Обзор методов прогнозирования дефектов программного обеспечения // Программные продукты, системы и алгоритмы. 2019. № 1. С. 1–4.

17. Демидов Р. А. Поиск уязвимостей в машинном коде с помощью методов глубокого обучения // Материалы IV межрегиональной научно-практической конференции «Перспективные направления развития отечественных информационных технологий материалы». 2018. С. 237–238.

18. Захаров А. А., Костюков А. Д. Использование алгоритмов машинного обучения в обнаружении вредоносного кода // Материалы международной научно-практической конференции «Проблемы социально-экономического и культурного развития России в условиях преодоления кризиса». 2017. С. 103–111.

References

1. Izrailov K. Analysis of the Security State of Software // II International Scientific and Technical and Methodological Conference «Actual Problems of Infotelecommunications in Science and Education». – SPb.: SUT, 2013. – pp. 874–877.

2. Buinevich M.V., Izrailov K.E., Mostovich D.I., Yaroshenko A.Yu. Problematic Issues of Neutralization of Vulnerabilities in a Software Code of Telecommunication Devices // Problems of Technosphere Risk Management. 2016. No 3 (39). pp. 81–89.

3. Buynevich M. V., Vasilyeva I. N., Vorobyev T. M., Gnidenko I. G., Egorova I. V., Enikeyeva L. A. i dr. Zashchita informatsii v kompyuternykh sistemakh: monografiya. – SPb.: SPGEU. 2017. 163 s.

4. Buynevich M. V., Izrailov K. E., Mostovich D. I. Sravnitelnyy analiz podkhodov k poisku uyazvimostey v programmnom kode // Aktualnyye problemy infotelekommunikatsiy v nauke i obrazovanii (APINO-2016): sbornik nauchnykh statey V Mezhdunarodnoy nauchno-tekhnicheskoy i nauchno-metodicheskoy konferentsii. 2016. V. 1. pp. 256–260.

5. Izrailov K. E. Poisk uyazvimostey v razlichnykh predstavleniyakh mashinnogo koda // Informatsionnaya bezopasnost regionov Rossii (IBRR-2015): materialy IKh Sankt-Peterburgskoy mezhregionalnoy konferentsii. 2015. S. 157.

6. Izrailov K. E., Gololobov N. V., Kraskin G. A. Method of Malware Analysis Based on Fuzzy Hash // Informatizatsiya i svyaz. 2019. № 2. S. 36–44.

7. Izrailov K. Model of Forecasting the Telecommunication System Threats on the Basis of the Artificial Neural Network // Вест Vestnik INZhEKONa. Seriya: Tekhnicheskiye nauki. 2012. № 8(59). S. 150–153.

8. Biryukov A., Izrailov K. The comparative analysis of models of threats of information security for the benefit of applicability for multi-stage schemes of the attacks // VI International Scientific and Technical and Methodological Conference «Actual Problems of Infotelecom-munications in Science and Education». 2017. pp. 108–112.

9. Balueva A., Desnitsky V. Approach to Detection of Denial-of-Sleep Attacks in Cyber-Physical Systems on the Base of Machine-Training Methods. // VIII International Scientific and Technical and Methodological Conference «Actual Problems of Infotelecommunications in Science and Education». 2019. pp. 97–103.

10. Buinevich M., Izrailov K., Vladyko A. The life cycle of vulnerabilities in the representations of software for telecommunication devices // 18th International Conference On Advanced Communications Technology (ICACT-2016). 2016. PP. 430–435.

11. Buinevich M., Izrailov K., Vladyko A. Metric of vulnerability at the base of the life cycle of software representations // 20th International Conference on Advanced Communication Technology (ICACT 2018). 2018. PP. 1–8.

12. Izrailov K. E. Arkhitekturnyye uyazvimosti programmnogo obespecheniya // Shestoy nauchnyy kongress studentov i aspirantov SPbGIEU (INZhEKON-2013): sbornik tezisov dokladov nauchno-prakticheskoy konferentsii fakulteta informatsionnykh sistem v ekonomike i upravlenii «Infokommunikatsionnyye tekhnologii i matematicheskiye metody». 2013. S. 35.

13. Buinevich M. V., Izrailov K. E., Scherbakov O. V. Structural Model of Machine Code Specialized for Search for Software Vulnerabilities of the Automated Control Systems // Problems of Technosphere Risk Management. 2014. N 3 (31). pp. 68–74.

14. Buinevich M. V., Izrailov K. E., Scherbakov O. V. Model of Machine Code Specialized for Vulnerabilities Search // Vestnik Voronezhskogo instituta GPS MChS Rossii. 2014. № 2(11). S. 46–51.

15. Izrailov K. E. Rassmotreniye predstavleniy koda programm s pozitsiy metadannykh // Fundamentalnyye issledovaniya i innovatsii v natsionalnykh issledovatelskikh universitetakh: materialy Vserossiyskoy nauchno-metodicheskoy konferentsii. 2012. T. 5. S. 176–180.

16. Belov Yu. S., Yukhimenko N. V. Obzor metodov prognozirovaniya defektov programmnogo obespecheniya // Software Journal: Theory and Applications. 2019. № 1. S. 1–4.

17. Demidov R. A. Poisk uyazvimostey v mashinnom kode s pomoshchyu metodov glubokogo obucheniya // Materialy IV mezhregionalnoy nauchno-prakticheskoy konferentsii «Perspektivnyye napravleniya razvitiya otechestvennykh informatsionnykh tekhnologiy materialy». 2018. S. 237–238.

18. Zakharov A. A., Kostyukov A. D. Ispolzovaniye algoritmov mashinnogo obucheniya v obnaruzhenii vredonosnogo koda // Materialy mezhduna-rodnoy nauchno-prakticheskoy konferentsii «Problemy sotsialno-ekonomicheskogo i kulturnogo razvitiya Rossii v usloviyakh preodoleniya krizisa». 2017. S. 103–111.

Буйневич Михаил Викторович – доктор технических наук, профессор, СПбГУТ, Санкт-Петербург, 193232, Российская Федерация; профессор, Санкт-Петербургский университет ГПС МЧС России, 196105, Российская Федерация, bmv1958@yandex.ru

Жуковская Полина Евгеньевич – студентка, СПбГУТ, Санкт-Петербург, 193232, Российская Федерация, zhukovskaua98@mail.ru

Израилов Константин Евгеньевич – кандидат технических наук, доцент, СПбГУТ, Санкт-Петербург, 193232, Российская Федерация, Konstantin.Izrailov@mail.ru

Покусов Виктор Владимирович – аспирант, Санкт-Петербургский университет ГПС МЧС России, 196105, Российская Федерация, v@victor.kz

Buinevich Mikhail – Doctor of Engineering Sciences, Full Professor, SUT, St. Petersburg, 193232, Russian Federation; Professor Saint-Petersburg University of State Fire Service of EMERCOM of Russia, 196105, Russian Federation, bmv1958@yandex.ru

Zhukovskaya Polina – Student, SUT, St. Petersburg, 193232, Russian Federation, zhukovskaua98@mail.ru

Izrailov Konstantin – Candidate of Engineering Sciences, Associate Professor, SUT, St. Petersburg, 193232, Russian Federation, Konstantin.Izrailov@mail.ru

Pokussov Viktor – Postgraduate, Saint-Petersburg University of State Fire Service of EMERCOM of Russia, 196105, Russian Federation, v@victor.kz